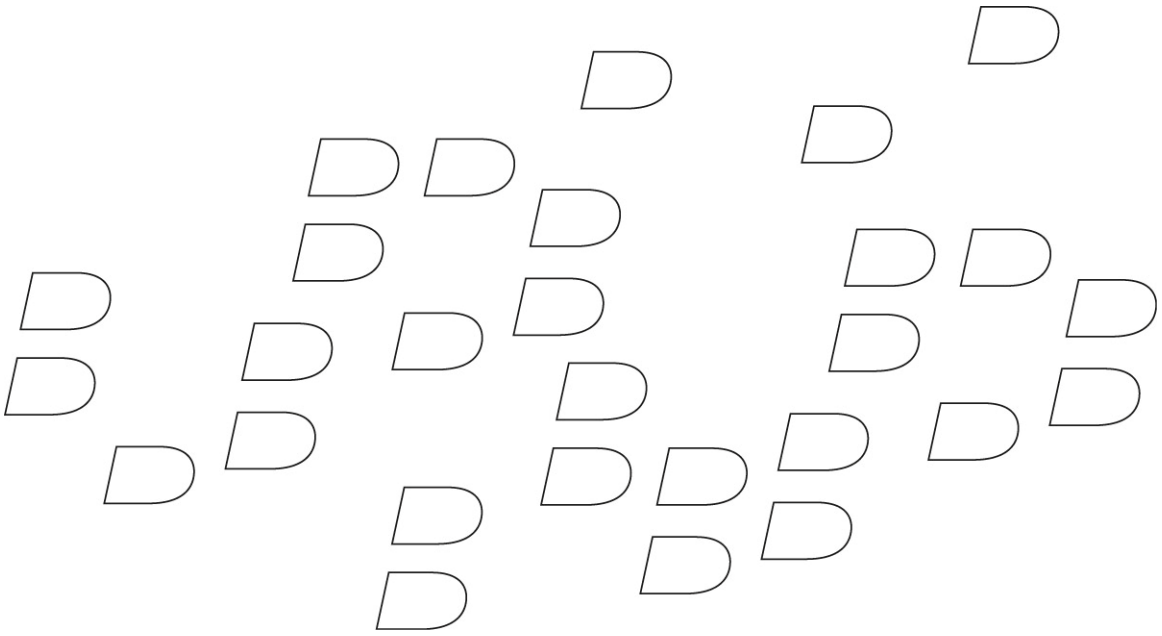

BlackBerry Java Development Environment

Version 4.3.0

Release Notes and Known Issues List

© 2007 Research In Motion Limited. All rights reserved. www.blackberry.com

NOTE

This document is provided for informational purposes only, and does not constitute a binding legal document unless specifically incorporated by reference into a binding legal agreement between you and Research In Motion (RIM). In the event that you enter into a binding legal agreement with RIM, all provisions contained in such binding legal agreement shall apply, regardless of whether such provisions conflict with information contained herein.

©2007 Research In Motion Limited. All Rights Reserved. The BlackBerry and RIM families of related marks, images, and symbols are the exclusive properties of Research In Motion Limited. RIM, Research In Motion, BlackBerry, “Always On, Always Connected” and the “envelope in motion” symbol are registered with the U.S. Patent and Trademark Office and may be pending or registered in other countries.

Java and J2ME are registered trademarks of Sun Microsystems, Inc. Microsoft Vista, DirectX, and Windows are trademarks of Microsoft Corporation. Intel and Pentium are trademarks of Intel Corporation. Bluetooth is a trademark of Bluetooth SIG. All other brands, product names, company names, trademarks and service marks are the properties of their respective owners.

The BlackBerry device and/or associated software are protected by copyright, international treaties, and various patents, including one or more of the following U.S. patents: 6,278,442; 6,271,605; 6,219,694; 6,075,470; 6,073,318; D445,428; D433,460; D416,256. Other patents are registered or pending in various countries around the world. Visit www.rim.com/patents.shtml for a current list of RIM [as hereinafter defined] patents.

This document is provided “as is” and Research In Motion Limited (RIM) assumes no responsibility for any typographical, technical, or other inaccuracies in this document. RIM reserves the right to periodically change information that is contained in this document; however, RIM makes no commitment to provide any such changes, updates, enhancements, or other additions to this document to you in a timely manner or at all. RIM MAKES NO REPRESENTATIONS, WARRANTIES, CONDITIONS, OR COVENANTS, EITHER EXPRESS OR IMPLIED (INCLUDING, WITHOUT LIMITATION, ANY EXPRESS OR IMPLIED WARRANTIES OR CONDITIONS OF FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, MERCHANTABILITY, DURABILITY, TITLE, OR RELATED TO THE PERFORMANCE OR NON-PERFORMANCE OF ANY SOFTWARE REFERENCED HEREIN, OR PERFORMANCE OF ANY SERVICES REFERENCED HEREIN). IN CONNECTION WITH YOUR USE OF THIS DOCUMENTATION, NEITHER RIM NOR THEIR RESPECTIVE DIRECTORS, OFFICERS, EMPLOYEES, OR CONSULTANTS SHALL BE LIABLE TO YOU FOR ANY DAMAGES WHATSOEVER BE THEY DIRECT, ECONOMIC, COMMERCIAL, SPECIAL, CONSEQUENTIAL, INCIDENTAL, EXEMPLARY, OR INDIRECT DAMAGES, EVEN IF RIM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, INCLUDING, WITHOUT LIMITATION, LOSS OF BUSINESS REVENUE OR EARNINGS, LOST DATA, DAMAGES CAUSED BY DELAYS, LOST PROFITS, OR A FAILURE TO REALIZE EXPECTED SAVINGS.

This document might contain references to third-party sources of information and/or third-party web sites (“Third-Party Information”). RIM does not control, and is not responsible for, any Third-Party Information, including, without limitation, the content, accuracy, copyright compliance, legality, decency, links, or any other aspect of Third-Party Information. The inclusion of Third-Party Information in this document does not imply endorsement by RIM of the third party in any way. Any dealings with third parties, including, without limitation, compliance with applicable licenses, and terms and conditions are solely between you and the third party. RIM shall not be responsible or liable for any part of such dealings.

Send us your comments on release notes and known issues lists:
https://www.blackberry.com/DocsFeedback.

12077369

About the BlackBerry JDE

The BlackBerry® Java® Development Environment is a fully integrated development environment and simulation tool for building Java 2 Platform, Micro Edition applications for Java-based BlackBerry® devices.

It is a Mobile Information Device Profile (MIDP) compliant J2ME™ environment for developers who wish to maintain seamless portability in their wireless applications.

Use the BlackBerry JDE to build applications that take advantage of the unique features of the BlackBerry device, such as desktop synchronization applications, GPS applications, and applications that push content proactively to BlackBerry devices in environments that use the BlackBerry Enterprise Server. The BlackBerry JDE provides ways for BlackBerry device applications to establish network connections to servers on the Internet or the corporate intranet.

Each version of the BlackBerry JDE comes with the BlackBerry device simulators that were available when that version of the BlackBerry JDE was made public. You can download additional BlackBerry device simulators from the BlackBerry Developer Zone web site:

http://www.blackberry.com/developers/index.shtml.

Documentation

The following documents for the BlacKBerry JDE Version 4.3.0 are available on the BlackBerry Developer Zone web site: http://www.blackberry.com/developers.

- BlackBerry Java Development Environment Development Guide
- BlackBerry Java Development Environment Fundamentals Guide
- Multimedia Development Guide
- GPS and BlackBerry Maps Development Guide
- BlackBerry Signature Tool Developer Guide
- Cryptographic Smart Card Driver Development Guide
- BlackBerry Email and MDS Services Simulator Package release notes
- BlackBerry JDE Component Package release notes
- Release Notes and Known Issues List

New content for the BlackBerry JDE Version 4.3.0

Document	Location in document	Topic
BlackBerry Java Development Environment Development Guide	Chapter 2	create a player that displays a video in a field let an application's screen display content created in Plazmic Media Engine Version 4.2. start the media player with no content start the media player with content

© 2007 Research In Motion Limited. All rights reserved.

Document	Location in document	Topic
		create a player for streaming media
	Chapter 8	working with Wi-Fi connections on a BlackBerry device
	Chapter 14	retrieve the phone number of a BlackBerry device
	Appendix D	accessing application attribute properties from a .jad file
BlackBerry Signature Tool Development Guide	Appendix A	information on the -p attribute
Cryptographic Smart Card Driver Development Guide		a new guide that focuses on how to create cryptographic smart card drivers
GPS and BlackBerry Maps Guide	Chapter 3	information on running the BlackBerryMapsDemo project and related maps samples in the 4.3.0 Integrated Development Environment

Getting started with the BlackBerry JDE

You can start the BlackBerry® Java Development Environment in one of two ways:

1. From the Start menu, select:

   Programs > **Research In Motion** > **BlackBerry JDE 4.3.0** > **JDE**

2. Invoke ide.bat, located in the bin subdirectory of the BlackBerry JDE installation directory.

Internationalization

The BlackBerry® Java Development Environment contains tools that assist with internationalization (i18n) efforts. It is important to follow naming conventions. All the sample applications include the use of i18n. Please review those samples for more information.

Note: The .rrc and .rrh files must exist in the same directory for successful compilation.

Note: You should define resource keys in uppercase letters only.

See the *BlackBerry Java Development Environment Development Guide* or the *BlackBerry Integrated Development Environment Online Help* for more information about localization.

© 2007 Research In Motion Limited. All rights reserved.

Code samples

A sample BlackBerry® JDE workspace exists in the samples subdirectory of the BlackBerry JDE installation directory. It contains sample programs that demonstrate how to design and build Java applications for BlackBerry devices.

The following sample applications include server components. Invoke the run.bat script in the specified folder to start the server component:

- 3.1 The httppushdemo sample includes a server component at samples/com/rim/samples/server/httppushdemo
- 3.2 The socketdemo sample includes a server component at samples/com/rim/samples/server/socketdemo
- 3.3 The browserpushdemo sample includes a server component at samples/com/rim/samples/server/browserpushdemo
- 3.4 The smsdemo sample includes a server component at samples/com/rim/samples/server/smsdemo
- 3.5 The BrowserMultipartPushDemo sample includes a server component at samples/com/rim/samples/server/browsermultipartpushdemo
- 3.6 The GPS Demo sample includes a server component at samples/com/rim/samples/server/gpsdemo

BlackBerry JDE Version 4.3.0 system requirements

Your system must meet the following requirements to support the BlackBerry® Java™ Development Environment:

- Microsoft® Vista™ or Windows XP
- Intel® Pentium® III processor or compatible (800 MHz or higher)
- 256 MB RAM
- 100 MB free hard disk space
- Java® SE Development Kit (JDK) Version 5 or Version 6 (download from the Sun Microsystems web site). The <bin> directory of the Java SDK must be part of the system's PATH variable.
- Microsoft DirectX® 8.0 or greater (for the BlackBerry device simulator)

Functional changes

This section lists functional changes for BlackBerry® Java Development Environment (JDE) Version 4.3.0.

Note: The JSRs listed in the following table are in the process of being tested and implemented by RIM.

Take a screen shot of the BlackBerry device screen	You can invoke the screenshot(Bitmap bitmap) and screenshot(Bitmap bitmap, int x, int y, int width, int height) methods of the net.rim.device.api.system.Display class to take a screen shot of the BlackBerry® device display or a portion of the display. The screen shot data is saved into a bitmap object. Example uses for this functionality include technical support and troubleshooting activities. After saving the screenshot, you can encode it with PNGEncodedImage.encode or JPEGEncodedImage.encode to create a byte representation that can be written to a file.
BlackBerry Maps	In the BlackBerry Java Development Environment Version 4.3.0, the code samples referred to in the *GPS and BlackBerry Maps Development Guide* are now combined

© 2007 Research In Motion Limited. All rights reserved.

samples	into a single project. The BlackBerryMapsDemo project now includes the InvokeContact.java, InvokeDefault.java, InvokeLocationDocument.java and InvokeMapView.java sample files as separate MainScreen classes. To launch a BlackBerry Maps sample: 1. From the BlackBerry Integrated Development Environment, run the BlackBerryMapsDemo project. 2. From the application menu, select a menu item to launch one of the BlackBerry Maps samples.
ActiveTextFields application	The ActiveTextFields application demonstrates the functionality of the ActiveRichTextField, ActiveAutoTextEditField, and PatternRepository classes. When the BlackBerry® device starts, the ActiveTextFields application registers a regular expression with the string pattern repository and registers menu items for the regular expression. If the regular expression pattern occurs in an active text field in a BlackBerry device application, the text is underlined and specific context menu items are made available. If a BlackBerry device user selects a menu item, the ActiveTextFields application is invoked and a handler screen displays information that corresponds to the matched pattern and the menu item.
Standard access from J2ME to web services	RIM now supports the J2ME™ Web Services API (JSR 172) that provides access from J2ME to web services. You can create BlackBerry® device applications that let BlackBerry devices act as web service clients.
Support for internationalization of MIDlets	RIM's support for the Mobile Internationalization API (JSR 238) allows you to create localizable application resources outside of application source code. At run time, an application can access, select, and use the localizable resources that match the BlackBerry® device's locale.
Accessing volume key events from a MIDlet	MIDlets can use the Canvas.keyPressed() method to receive values that represent volume key events. ▪ The keyCode integer value for the Volume Up key is -150. ▪ The keyCode integer value for the Volume Down key is -151.
Enhanced support for multi-media and web content management	RIM now supports the Content Handler API (JSR 211). You can use this API to create BlackBerry® device applications that can register as content handler applications and use Uniform Resource Identifiers based on MIME-type or scheme.
Auto-detection of debug information by the BlackBerry IDE	If you have BlackBerry® Device Simulator Package Version 4.2.2 or higher installed on your computer, when you attach the BlackBerry Integrated Development Environment debugger to a BlackBerry device, the BlackBerry IDE will automatically detect the debug information.
Support for sending and receiving MMS	RIM now supports Wireless Messaging API 2.0 (JSR 205). You can use this API to create BlackBerry® device applications that send and receive messages that include images, sound, video, and text.
Registering string patterns for active fields	A BlackBerry® device application can register multiple menu items for multiple string patterns for a single string pattern. The string pattern matching routine and menu item creation logic runs only within the process of the BlackBerry device application. See the net.rim.blackberry.api.stringpattern package in the API Reference for more information.
Launching the media player	A BlackBerry® device application can use the javax.microedition.content and net.rim.device.api.content packages to launch the media player with or without content.
Streaming media in a media player.	You can create a media player that uses a Real Time Streaming Protocol (RTSP) locator string to play streaming media.
Accessing and	A BlackBerry® device application can set and get the time when a task reminder

© 2007 Research In Motion Limited. All rights reserved.

setting task reminder information	occurs using the BlackBerryToDo.REMINDER field.
Adding birthday and anniversary information for a PIM contact.	A BlackBerry® device application can use the javax.microedition.pim.Contact.BIRTHDAY field or the net.rim.blackberry.api.pdap.BlackBerryContact.ANNIVERSARY field to add birthday and anniversary information for a PIM contact.
Work with the lock timeout value of the BlackBerry device	A BlackBerry® device application can invoke net.rim.device.api.system.DeviceInfo.canResetIdleTime() to determine if it can extend the BlackBerry device past the lock timeout. A BlackBerry device application can invoke net.rim.device.api.system.DeviceInfo.getLockTimeout() to retrieve the lock timeout value of the BlackBerry device.
Work with multiple BlackBerry Device Simulator packages and profiles	You can use fledgelauncher.exe to create and use device profiles using BlackBerry Device Simulators from the BlackBerry JDE and BlackBerry simulator packages. The fledgelauncher application, included with the BlackBerry Device Simulator, provides a GUI for running the BlackBerry Device Simulator using command line options.
Retrieve the version of the BlackBerry Device Software.	A BlackBerry® device application can invoke the getSoftwareVersion() method of the DeviceInfo class to retrieve the version number of the BlackBerry Device Software on a BlackBerry device.
Enhanced autotext support in a text field.	A BlackBerry® device application can use the following new fields of the AutoTextField class: ▪ AUTOCAP_OFF - Do not allow automatic capitalization based on preceding punctuation. ▪ AUTOPERIOD_OFF - Do not allow automatic period insertion after two spaces. ▪ AUTOREPLACE_OFF - Do not allow autotext substitution on this field.
Receive notifications when the Wi-Fi radio is connected or disconnected to an access point.	A BlackBerry® device application can use the net.rim.device.api.system.WLANConnectionListener interface and the net.rim.device.api.system.WLANInfo.addListener() method to receive notifications when the Wi-Fi radio on a BlackBerry device is connected or disconnected to an access point.
Determine status information for the Wi-Fi radio on a BlackBerry device.	A BlackBerry® device application can use the net.rim.device.api.system.WLANInfo class and the net.rim.device.api.system.WLANInfo.WLANAPInfo class to determine status information about the Wi-Fi radio on a BlackBerry device. An application can determine if the Wi-Fi radio is connected or not connected to an access point. If connected to an access point, a BlackBerry device application can obtain status information such as the data rate of the connection, the radio band used, the SSID of the connected access point, and the name of the Wi-Fi profile in use.
Accessing application attribute properties from a BlackBerry MDS Java Application .jad file	After a BlackBerry® device user uses the .jad file of a BlackBerry device application to download the application on to a BlackBerry device, the application can use the CodeModuleGroup to access the name-value pair information in the application's .jad file.
TicTacToe sample	You can now play a two-player tic tac toe game against a BlackBerry® Messenger contact.
Display a video in a UI field	You can invoke the initDisplayMode method of the VideoControl class to create an instance of a Field object to display video in the UI of a BlackBerry® device

© 2007 Research In Motion Limited. All rights reserved.

application. See the *API Reference* for more information.

Known issues

This section lists known issues for the BlackBerry Java Development Environment (JDE) Version 4.3.0.

Installing and using the BlackBerry JDE on Microsoft® Vista™

To install and use the BlackBerry® JDE Version 4.3.0 on a computer with Microsoft® Vista™, you must have administrative permissions on the computer.

Using the BlackBerry Email Simulator with Microsoft Outlook to send and receive emails

In Microsoft® Outlook®, if you send an email to an account on a BlackBerry® Smartphone Simulator that uses the BlackBerry Email Simulator as the POP3 and SMTP server, Microsoft Outlook does not send the email. Microsoft Outlook displays the following response: “error code 503 ‘No sender’”.

Impact: If you use Microsoft Outlook to send an email to an account that uses the BlackBerry Email Simulator as the POP3 and SMTP server, Microsoft Outlook does not send the email.

Workaround: Modify the BlackBerry Smartphone Simulator account in Microsoft Outlook to use the BlackBerry MDS Simulator instead of the BlackBerry Email Simulator:

1. Open the rimpublic.property file located in C:\Program Files\Research In Motion\BlackBerry JDE x.x.x\MDS.

2. Modify the rimpublic.property file as follows:

```
[Email]
Email.mode              =standalone
#Email.mode             =connect
#Email.mode             =none
#Common settings for both standalone and connect mode
Email.personal    =Test User
Email.address     =tester@device.test
Email.deviceId    =2100000a

#Connect mode settings
Email.pop3Server  =localhost
Email.smtpServer  =localhost
Email.smtpPort    =25
Email.pop3Port    =110
Email.userId =tester
Email.password        =
Email.pollInterval=30
```

3. In Microsoft Outlook, set up an account that uses the BlackBerry MDS Simulator as the POP3 and SMTP server.

    1. From the **Start** menu, click **Settings** > **Control Panel** > **Mail**.

© 2007 Research In Motion Limited. All rights reserved.

2. Click **Show Profiles**.
3. Click **Add**.
4. In the **Profile Name** field, type a name.
5. Click **OK**.
6. Click **Next**.
7. Make sure **POP3** is selected.
8. Click **Next**.
9. In the **Your Name** field, type a name.
10. In the **E-mail Address** field, type an email address.
11. In the **User Name** field, type **tester**.
12. In the **Incoming mail server (POP3)** field, type **localhost**.
13. In the **Outgoing mail server (SMTP)** field, type **localhost**.
14. Click **Next**.
15. Click **Finish**.
16. In the Mail window, make sure that **Prompt for a profile to be used** is selected
17. Restart Microsoft Outlook.
18. In the **Choose Profile** dialog, in the **Profile Name** field, select the profile that you created.
19. Click **OK**.

4. Start the BlackBerry MDS Simulator.

5. In Microsoft Outlook, send an email to the account on the BlackBerry Smartphone Simulator.

OTASyncDemo

When you run the OTASyncDemo sample, after you activate a BlackBerry Enterprise Server user's BlackBerry device with the Sync Server SDK, any subsequent activations generate an IO exception that prevent activation from completing.

Impact: After activating a BlackBerry Enterprise Server user and clearing the file system on the BlackBerry Smartphone Simulator, the sample will not restore contacts saved by the current BlackBerry Enterprise Server user.

Workaround: None

Viewing processes in the BlackBerry IDE

If you are debugging an application and you attempt to view information on processes that are running on the BlackBerry® Smartphone Simulator when the debugger is not at a breakpoint, the BlackBerry® IDE does not display process information and stops responding.

Impact: If you debug an application in the BlackBerry IDE and you click **View** > **Processes** when the debugger is not at a breakpoint, the BlackBerry IDE does not display process information and stops responding to user input.

Workaround: To close the BlackBerry IDE, use the Task Manager on your computer.

Using Eclipse with the BlackBerry JDE Component Package

To use the BlackBerry® Java® Development Environment Component Package with the Eclipse development environment, you must set Eclipse to use a Java compiler compliance level of 1.4.

Impact: When you compile an application, errors might occur when Eclipse attempts to use a Java compiler compliance level other than 1.4.

Workaround: Perform the following steps:

1. Open Eclipse.
2. In the Project menu, click **Properties**.
3. Select **Java Compiler**.
4. In the **Compiler compliance level** field, select **1.4**.
5. Click **Apply**.
6. Click **OK**.

Registering code signatures

The BlackBerry® Signature tool included with the BlackBerry JDE Version 4.3.0 uses a different private key (.csk) file format than previous versions of the BlackBerry Signature

© 2007 Research In Motion Limited. All rights reserved.

	tool. The first time you use the new version of the BlackBerry Signature tool with an old private key to register code signatures, the Signature tool prompts you for the private key password and updates the format of the private key file. After this process completes, you can only use the BlackBerry Signature tool included with the BlackBerry JDE Version 4.3.0. **Impact:** Once you use the BlackBerry Signature Tool included with the BlackBerry JDE Version 4.3.0, you cannot use an older version of the Signature tool. **Workaround:** None.
Opening a connection using the jar:// protocol	You cannot use the jar:// protocol with the Connector.open method, for example, Connector.open("jar://some_loc"). **Impact:** You cannot open a connection using the jar:// protocol. **Workaround:** None.
SDR132238	When you run the BlackBerry Device Simulator from the BlackBerry Integrated Development Environment on a computer that runs Microsoft® Windows Vista™, the BlackBerry Device Simulator does not create the flash memory files for the device profile. **Impact**: When you run the BlackBerry Device Simulator from the BlackBerry IDE on a computer that runs Microsoft® Windows Vista™, the BlackBerry Device Simulator does not generate .dmp files in the C:\Program Files\Research In Motion\BlackBerry JDE 4.3.0\simulator directory. **Workaround**: None

BlackBerry Java Development Environment Version 4.3.0 - new and deprecated methods

New methods

- java.lang.IndexOutOfBoundsException.getMessage()
- java.rmi.MarshalException.MarshalException(java.lang.String)
- java.rmi.MarshalException.MarshalException(java.lang.String, java.lang.Exception)
- java.rmi.ServerException.ServerException(java.lang.String)
- java.rmi.ServerException.ServerException(java.lang.String, java.lang.Exception)
- javax.microedition.content.ActionNameMap.ActionNameMap(java.lang.String[], java.lang.String[], java.lang.String)
- javax.microedition.content.ActionNameMap.getAction(int)
- javax.microedition.content.ActionNameMap.getAction(java.lang.String)
- javax.microedition.content.ActionNameMap.getActionName(int)
- javax.microedition.content.ActionNameMap.getActionName(java.lang.String)
- javax.microedition.content.ActionNameMap.getLocale()
- javax.microedition.content.ActionNameMap.size()
- javax.microedition.content.ContentHandler.ACTION_EDIT
- javax.microedition.content.ContentHandler.ACTION_EXECUTE

© 2007 Research In Motion Limited. All rights reserved.

- javax.microedition.content.ContentHandler.ACTION_INSTALL
- javax.microedition.content.ContentHandler.ACTION_NEW
- javax.microedition.content.ContentHandler.ACTION_OPEN
- javax.microedition.content.ContentHandler.ACTION_PRINT
- javax.microedition.content.ContentHandler.ACTION_SAVE
- javax.microedition.content.ContentHandler.ACTION_SELECT
- javax.microedition.content.ContentHandler.ACTION_SEND
- javax.microedition.content.ContentHandler.ACTION_STOP
- javax.microedition.content.ContentHandler.UNIVERSAL_TYPE
- javax.microedition.content.ContentHandler.getAction(int)
- javax.microedition.content.ContentHandler.getActionCount()
- javax.microedition.content.ContentHandler.getActionNameMap()
- javax.microedition.content.ContentHandler.getActionNameMap(int)
- javax.microedition.content.ContentHandler.getActionNameMap(java.lang.String)
- javax.microedition.content.ContentHandler.getActionNameMapCount()
- javax.microedition.content.ContentHandler.getAppName()
- javax.microedition.content.ContentHandler.getAuthority()
- javax.microedition.content.ContentHandler.getID()
- javax.microedition.content.ContentHandler.getSuffix(int)
- javax.microedition.content.ContentHandler.getSuffixCount()
- javax.microedition.content.ContentHandler.getType(int)
- javax.microedition.content.ContentHandler.getTypeCount()
- javax.microedition.content.ContentHandler.getVersion()
- javax.microedition.content.ContentHandler.hasAction(java.lang.String)
- javax.microedition.content.ContentHandler.hasSuffix(java.lang.String)
- javax.microedition.content.ContentHandler.hasType(java.lang.String)
- javax.microedition.content.ContentHandlerException.AMBIGUOUS
- javax.microedition.content.ContentHandlerException.NO_REGISTERED_HANDLER
- javax.microedition.content.ContentHandlerException.TYPE_UNKNOWN
- javax.microedition.content.ContentHandlerException.ContentHandlerException(java.lang.String, int)
- javax.microedition.content.ContentHandlerException.getErrorCode()
- javax.microedition.content.ContentHandlerServer.accessAllowedCount()
- javax.microedition.content.ContentHandlerServer.cancelGetRequest()
- javax.microedition.content.ContentHandlerServer.finish(javax.microedition.content.Invocation, int)
- javax.microedition.content.ContentHandlerServer.getAccessAllowed(int)
- javax.microedition.content.ContentHandlerServer.getRequest(boolean)
- javax.microedition.content.ContentHandlerServer.isAccessAllowed(java.lang.String)

© 2007 Research In Motion Limited. All rights reserved.

- javax.microedition.content.ContentHandlerServer.setListener(javax.microedition.content.RequestListener)
- javax.microedition.content.Invocation.ACTIVE
- javax.microedition.content.Invocation.CANCELLED
- javax.microedition.content.Invocation.ERROR
- javax.microedition.content.Invocation.HOLD
- javax.microedition.content.Invocation.INIT
- javax.microedition.content.Invocation.INITIATED
- javax.microedition.content.Invocation.OK
- javax.microedition.content.Invocation.WAITING
- javax.microedition.content.Invocation.Invocation()
- javax.microedition.content.Invocation.Invocation(java.lang.String)
- javax.microedition.content.Invocation.Invocation(java.lang.String, java.lang.String)
- javax.microedition.content.Invocation.Invocation(java.lang.String, java.lang.String, java.lang.String)
- javax.microedition.content.Invocation.Invocation(java.lang.String, java.lang.String, java.lang.String, boolean, java.lang.String)
- javax.microedition.content.Invocation.findType()
- javax.microedition.content.Invocation.getAction()
- javax.microedition.content.Invocation.getArgs()
- javax.microedition.content.Invocation.getData()
- javax.microedition.content.Invocation.getID()
- javax.microedition.content.Invocation.getInvokingAppName()
- javax.microedition.content.Invocation.getInvokingAuthority()
- javax.microedition.content.Invocation.getInvokingID()
- javax.microedition.content.Invocation.getPrevious()
- javax.microedition.content.Invocation.getResponseRequired()
- javax.microedition.content.Invocation.getStatus()
- javax.microedition.content.Invocation.getType()
- javax.microedition.content.Invocation.getURL()
- javax.microedition.content.Invocation.open(boolean)
- javax.microedition.content.Invocation.setAction(java.lang.String)
- javax.microedition.content.Invocation.setArgs(java.lang.String[])
- javax.microedition.content.Invocation.setCredentials(java.lang.String, char[])
- javax.microedition.content.Invocation.setData(byte[])
- javax.microedition.content.Invocation.setID(java.lang.String)
- javax.microedition.content.Invocation.setResponseRequired(boolean)
- javax.microedition.content.Invocation.setType(java.lang.String)
- javax.microedition.content.Invocation.setURL(java.lang.String)

© 2007 Research In Motion Limited. All rights reserved.

- javax.microedition.content.Registry.cancelGetResponse()
- javax.microedition.content.Registry.findHandler(javax.microedition.content.Invocation)
- javax.microedition.content.Registry.forAction(java.lang.String)
- javax.microedition.content.Registry.forID(java.lang.String, boolean)
- javax.microedition.content.Registry.forSuffix(java.lang.String)
- javax.microedition.content.Registry.forType(java.lang.String)
- javax.microedition.content.Registry.getActions()
- javax.microedition.content.Registry.getID()
- javax.microedition.content.Registry.getIDs()
- javax.microedition.content.Registry.getRegistry(java.lang.String)
- javax.microedition.content.Registry.getResponse(boolean)
- javax.microedition.content.Registry.getServer(java.lang.String)
- javax.microedition.content.Registry.getSuffixes()
- javax.microedition.content.Registry.getTypes()
- javax.microedition.content.Registry.invoke(javax.microedition.content.Invocation)
- javax.microedition.content.Registry.invoke(javax.microedition.content.Invocation, javax.microedition.content.Invocation)
- javax.microedition.content.Registry.register(java.lang.String, java.lang.String[], java.lang.String[], java.lang.String[], javax.microedition.content.ActionNameMap[], java.lang.String, java.lang.String[])
- javax.microedition.content.Registry.reinvoke(javax.microedition.content.Invocation)
- javax.microedition.content.Registry.setListener(javax.microedition.content.ResponseListener)
- javax.microedition.content.Registry.unregister(java.lang.String)
- javax.microedition.content.RequestListener.invocationRequestNotify(javax.microedition.content.ContentHandlerServer)
- javax.microedition.content.ResponseListener.invocationResponseNotify(javax.microedition.content.Registry)
- javax.microedition.global.Formatter.DATETIME_LONG
- javax.microedition.global.Formatter.DATETIME_SHORT
- javax.microedition.global.Formatter.DATE_LONG
- javax.microedition.global.Formatter.DATE_SHORT
- javax.microedition.global.Formatter.TIME_LONG
- javax.microedition.global.Formatter.TIME_SHORT
- javax.microedition.global.Formatter.Formatter()
- javax.microedition.global.Formatter.Formatter(java.lang.String)
- javax.microedition.global.Formatter.formatCurrency(double)
- javax.microedition.global.Formatter.formatCurrency(double, java.lang.String)
- javax.microedition.global.Formatter.formatDateTime(java.util.Calendar, int)
- javax.microedition.global.Formatter.formatMessage(java.lang.String, java.lang.String[])
- javax.microedition.global.Formatter.formatNumber(double)

© 2007 Research In Motion Limited. All rights reserved.

- javax.microedition.global.Formatter.formatNumber(double, int)
- javax.microedition.global.Formatter.formatNumber(long)
- javax.microedition.global.Formatter.formatPercentage(float, int)
- javax.microedition.global.Formatter.formatPercentage(long)
- javax.microedition.global.Formatter.getLocale()
- javax.microedition.global.Formatter.getSupportedLocales()
- javax.microedition.global.ResourceException.DATA_ERROR
- javax.microedition.global.ResourceException.METAFILE_NOT_FOUND
- javax.microedition.global.ResourceException.NO_RESOURCES_FOR_BASE_NAME
- javax.microedition.global.ResourceException.NO_SYSTEM_DEFAULT_LOCALE
- javax.microedition.global.ResourceException.RESOURCE_NOT_FOUND
- javax.microedition.global.ResourceException.UNKNOWN_ERROR
- javax.microedition.global.ResourceException.UNKNOWN_RESOURCE_TYPE
- javax.microedition.global.ResourceException.WRONG_RESOURCE_TYPE
- javax.microedition.global.ResourceException.ResourceException(int, java.lang.String)
- javax.microedition.global.ResourceException.getErrorCode()
- javax.microedition.global.ResourceManager.DEVICE
- javax.microedition.global.ResourceManager.getBaseName()
- javax.microedition.global.ResourceManager.getData(int)
- javax.microedition.global.ResourceManager.getLocale()
- javax.microedition.global.ResourceManager.getManager(java.lang.String)
- javax.microedition.global.ResourceManager.getManager(java.lang.String, java.lang.String)
- javax.microedition.global.ResourceManager.getManager(java.lang.String, java.lang.String[])
- javax.microedition.global.ResourceManager.getResource(int)
- javax.microedition.global.ResourceManager.getString(int)
- javax.microedition.global.ResourceManager.getSupportedLocales(java.lang.String)
- javax.microedition.global.ResourceManager.isCaching()
- javax.microedition.global.ResourceManager.isValidResourceID(int)
- javax.microedition.global.StringComparator.IDENTICAL
- javax.microedition.global.StringComparator.LEVEL1
- javax.microedition.global.StringComparator.LEVEL2
- javax.microedition.global.StringComparator.LEVEL3
- javax.microedition.global.StringComparator.StringComparator()
- javax.microedition.global.StringComparator.StringComparator(java.lang.String)
- javax.microedition.global.StringComparator.StringComparator(java.lang.String, int)
- javax.microedition.global.StringComparator.compare(java.lang.String, java.lang.String)
- javax.microedition.global.StringComparator.equals(java.lang.String, java.lang.String)
- javax.microedition.global.StringComparator.getLevel()

© 2007 Research In Motion Limited. All rights reserved.

- javax.microedition.global.StringComparator.getLocale()
- javax.microedition.global.StringComparator.getSupportedLocales()
- javax.microedition.global.UnsupportedLocaleException.UnsupportedLocaleException()
- javax.microedition.global.UnsupportedLocaleException.UnsupportedLocaleException(java.lang.String)
- javax.microedition.xml.rpc.ComplexType.elements
- javax.microedition.xml.rpc.ComplexType.value
- javax.microedition.xml.rpc.ComplexType.ComplexType()
- javax.microedition.xml.rpc.Element.UNBOUNDED
- javax.microedition.xml.rpc.Element.contentType
- javax.microedition.xml.rpc.Element.isArray
- javax.microedition.xml.rpc.Element.isNillable
- javax.microedition.xml.rpc.Element.isOptional
- javax.microedition.xml.rpc.Element.maxOccurs
- javax.microedition.xml.rpc.Element.minOccurs
- javax.microedition.xml.rpc.Element.name
- javax.microedition.xml.rpc.Element.value
- javax.microedition.xml.rpc.Element.Element(javax.xml.namespace.QName, javax.microedition.xml.rpc.Type)
- javax.microedition.xml.rpc.Element.Element(javax.xml.namespace.QName, javax.microedition.xml.rpc.Type, int, int, boolean)
- javax.microedition.xml.rpc.FaultDetailException.FaultDetailException(javax.xml.namespace.QName, java.lang.Object)
- javax.microedition.xml.rpc.FaultDetailException.getFaultDetail()
- javax.microedition.xml.rpc.FaultDetailException.getFaultDetailName()
- javax.microedition.xml.rpc.FaultDetailHandler.handleFault(javax.xml.namespace.QName)
- javax.microedition.xml.rpc.Operation.SOAPACTION_URI_PROPERTY
- javax.microedition.xml.rpc.Operation.Operation()
- javax.microedition.xml.rpc.Operation.invoke(java.lang.Object)
- javax.microedition.xml.rpc.Operation.newInstance(javax.xml.namespace.QName, javax.microedition.xml.rpc.Element, javax.microedition.xml.rpc.Element)
- javax.microedition.xml.rpc.Operation.newInstance(javax.xml.namespace.QName, javax.microedition.xml.rpc.Element, javax.microedition.xml.rpc.Element, javax.microedition.xml.rpc.FaultDetailHandler)
- javax.microedition.xml.rpc.Operation.setProperty(java.lang.String, java.lang.String)
- javax.microedition.xml.rpc.Type.BOOLEAN
- javax.microedition.xml.rpc.Type.BYTE
- javax.microedition.xml.rpc.Type.DOUBLE
- javax.microedition.xml.rpc.Type.FLOAT
- javax.microedition.xml.rpc.Type.INT

© 2007 Research In Motion Limited. All rights reserved.

- javax.microedition.xml.rpc.Type.LONG
- javax.microedition.xml.rpc.Type.SHORT
- javax.microedition.xml.rpc.Type.STRING
- javax.microedition.xml.rpc.Type.value
- javax.wireless.messaging.MessageConnection.MULTIPART_MESSAGE
- javax.wireless.messaging.MessagePart.MessagePart(byte[], int, int, java.lang.String, java.lang.String, java.lang.String, java.lang.String)
- javax.wireless.messaging.MessagePart.MessagePart(byte[], java.lang.String, java.lang.String, java.lang.String, java.lang.String)
- javax.wireless.messaging.MessagePart.MessagePart(java.io.InputStream, java.lang.String, java.lang.String, java.lang.String, java.lang.String)
- javax.wireless.messaging.MessagePart.getContent()
- javax.wireless.messaging.MessagePart.getContentAsStream()
- javax.wireless.messaging.MessagePart.getContentID()
- javax.wireless.messaging.MessagePart.getContentLocation()
- javax.wireless.messaging.MessagePart.getEncoding()
- javax.wireless.messaging.MessagePart.getLength()
- javax.wireless.messaging.MessagePart.getMIMEType()
- javax.wireless.messaging.MultipartMessage.addAddress(java.lang.String, java.lang.String)
- javax.wireless.messaging.MultipartMessage.addMessagePart(javax.wireless.messaging.MessagePart)
- javax.wireless.messaging.MultipartMessage.getAddress()
- javax.wireless.messaging.MultipartMessage.getAddresses(java.lang.String)
- javax.wireless.messaging.MultipartMessage.getHeader(java.lang.String)
- javax.wireless.messaging.MultipartMessage.getMessagePart(java.lang.String)
- javax.wireless.messaging.MultipartMessage.getMessageParts()
- javax.wireless.messaging.MultipartMessage.getStartContentId()
- javax.wireless.messaging.MultipartMessage.getSubject()
- javax.wireless.messaging.MultipartMessage.removeAddress(java.lang.String, java.lang.String)
- javax.wireless.messaging.MultipartMessage.removeAddresses()
- javax.wireless.messaging.MultipartMessage.removeAddresses(java.lang.String)
- javax.wireless.messaging.MultipartMessage.removeMessagePart(javax.wireless.messaging.MessagePart)
- javax.wireless.messaging.MultipartMessage.removeMessagePartId(java.lang.String)
- javax.wireless.messaging.MultipartMessage.removeMessagePartLocation(java.lang.String)
- javax.wireless.messaging.MultipartMessage.setAddress(java.lang.String)
- javax.wireless.messaging.MultipartMessage.setHeader(java.lang.String, java.lang.String)
- javax.wireless.messaging.MultipartMessage.setStartContentId(java.lang.String)
- javax.wireless.messaging.MultipartMessage.setSubject(java.lang.String)

© 2007 Research In Motion Limited. All rights reserved.

- javax.wireless.messaging.SizeExceededException.SizeExceededException(java.lang.String)
- javax.xml.namespace.QName.QName(java.lang.String)
- javax.xml.namespace.QName.QName(java.lang.String, java.lang.String)
- javax.xml.namespace.QName.QName(java.lang.String, java.lang.String, java.lang.String)
- javax.xml.namespace.QName.equals(java.lang.Object)
- javax.xml.namespace.QName.getLocalPart()
- javax.xml.namespace.QName.getNamespaceURI()
- javax.xml.namespace.QName.getPrefix()
- javax.xml.namespace.QName.hashCode()
- javax.xml.namespace.QName.toString()
- javax.xml.namespace.QName.valueOf(java.lang.String)
- javax.xml.parsers.FactoryConfigurationError.FactoryConfigurationError(java.lang.String)
- javax.xml.parsers.ParserConfigurationException.ParserConfigurationException(java.lang.String)
- javax.xml.parsers.SAXParser.SAXParser()
- javax.xml.parsers.SAXParser.isNamespaceAware()
- javax.xml.parsers.SAXParser.isValidating()
- javax.xml.parsers.SAXParser.parse(java.io.InputStream, org.xml.sax.helpers.DefaultHandler)
- javax.xml.parsers.SAXParser.parse(org.xml.sax.InputSource, org.xml.sax.helpers.DefaultHandler)
- javax.xml.parsers.SAXParserFactory.SAXParserFactory()
- javax.xml.parsers.SAXParserFactory.getFeature(java.lang.String)
- javax.xml.parsers.SAXParserFactory.isNamespaceAware()
- javax.xml.parsers.SAXParserFactory.isValidating()
- javax.xml.parsers.SAXParserFactory.newInstance()
- javax.xml.parsers.SAXParserFactory.newSAXParser()
- javax.xml.parsers.SAXParserFactory.setFeature(java.lang.String, boolean)
- javax.xml.parsers.SAXParserFactory.setNamespaceAware(boolean)
- javax.xml.parsers.SAXParserFactory.setValidating(boolean)
- javax.xml.rpc.JAXRPCException.JAXRPCException()
- javax.xml.rpc.JAXRPCException.JAXRPCException(java.lang.String)
- javax.xml.rpc.JAXRPCException.JAXRPCException(java.lang.String, java.lang.Throwable)
- javax.xml.rpc.JAXRPCException.JAXRPCException(java.lang.Throwable)
- javax.xml.rpc.JAXRPCException.getLinkedCause()
- javax.xml.rpc.NamespaceConstants.NSPREFIX_SCHEMA_XSD
- javax.xml.rpc.NamespaceConstants.NSPREFIX_SCHEMA_XSI
- javax.xml.rpc.NamespaceConstants.NSPREFIX_SOAP_ENVELOPE
- javax.xml.rpc.NamespaceConstants.NSURI_SCHEMA_XSD
- javax.xml.rpc.NamespaceConstants.NSURI_SCHEMA_XSI
- javax.xml.rpc.NamespaceConstants.NSURI_SOAP_ENCODING

© 2007 Research In Motion Limited. All rights reserved.

- javax.xml.rpc.NamespaceConstants.NSURI_SOAP_ENVELOPE
- javax.xml.rpc.NamespaceConstants.NSURI_SOAP_NEXT_ACTOR
- javax.xml.rpc.NamespaceConstants.NamespaceConstants()
- javax.xml.rpc.Stub.ENDPOINT_ADDRESS_PROPERTY
- javax.xml.rpc.Stub.PASSWORD_PROPERTY
- javax.xml.rpc.Stub.SESSION_MAINTAIN_PROPERTY
- javax.xml.rpc.Stub.USERNAME_PROPERTY
- javax.xml.rpc.Stub._getProperty(java.lang.String)
- javax.xml.rpc.Stub._setProperty(java.lang.String, java.lang.Object)
- net.rim.blackberry.api.invoke.CalculatorArguments.CalculatorArguments()
- net.rim.blackberry.api.menuitem.ApplicationMenuItemRepository.MENUITEM_MMS_EDIT
- net.rim.blackberry.api.menuitem.ApplicationMenuItemRepository.MENUITEM_MMS_VIEW
- net.rim.blackberry.api.pdap.BlackBerryContact.ANNIVERSARY
- net.rim.blackberry.api.pdap.BlackBerryToDo.REMINDER
- net.rim.blackberry.api.stringpattern.PatternRepository.PATTERN_TYPE_EXACT_MATCH
- net.rim.blackberry.api.stringpattern.PatternRepository.PATTERN_TYPE_REGULAR_EXPRESSION
- net.rim.blackberry.api.stringpattern.PatternRepository.addPattern(net.rim.device.api.system.ApplicationDescriptor, java.lang.String, int, net.rim.blackberry.api.menuitem.ApplicationMenuItem[])
- net.rim.blackberry.api.stringpattern.PatternRepository.removePatterns(net.rim.device.api.system.ApplicationDescriptor)
- net.rim.device.api.applicationcontrol.ApplicationPermissions.PERMISSION_INTER_PROCESS_COMMUNICATION
- net.rim.device.api.applicationcontrol.ApplicationPermissions.PERMISSION_MEDIA
- net.rim.device.api.applicationcontrol.ApplicationPermissions.PERMISSION_SCREEN_CAPTURE
- net.rim.device.api.applicationcontrol.ApplicationPermissions.PERMISSION_WIFI
- net.rim.device.api.content.BlackBerryContentHandler.ID_MEDIA_CONTENT_HANDLER
- net.rim.device.api.crypto.AESCryptoToken.initializeDecrypt(net.rim.device.api.crypto.CryptoTokenSymmetricKeyData, int, boolean)
- net.rim.device.api.crypto.AESCryptoToken.initializeEncrypt(net.rim.device.api.crypto.CryptoTokenSymmetricKeyData, int, boolean)
- net.rim.device.api.crypto.AESDecryptorEngine.BLOCK_LENGTH_DEFAULT
- net.rim.device.api.crypto.AESDecryptorEngine.AESDecryptorEngine(net.rim.device.api.crypto.AESKey, int, boolean)
- net.rim.device.api.crypto.AESEncryptorEngine.BLOCK_LENGTH_DEFAULT
- net.rim.device.api.crypto.AESEncryptorEngine.AESEncryptorEngine(net.rim.device.api.crypto.AESKey, int, boolean)
- net.rim.device.api.crypto.certificate.CertificateProperties.WEAK_CERTIFICATE_CHAIN_DIGEST
- net.rim.device.api.crypto.certificate.CertificateServerInfo.CertificateServerInfo(java.lang.String, int, int, java.lang.String, java.lang.String, int, int)
- net.rim.device.api.crypto.certificate.CertificateServerInfo.getConnectionType()

© 2007 Research In Motion Limited. All rights reserved.

- net.rim.device.api.crypto.certificate.CertificateServers.addServer(java.lang.String, int, java.lang.String, java.lang.String, int, int, int)
- net.rim.device.api.crypto.oid.OIDs.pbeWithSHAAnd128BitRC2_CBC
- net.rim.device.api.crypto.oid.OIDs.pbeWithSHAAnd128BitRC4
- net.rim.device.api.crypto.oid.OIDs.pbeWithSHAAnd2_KeyTripleDES_CBC
- net.rim.device.api.crypto.oid.OIDs.pbeWithSHAAnd3_KeyTripleDES_CBC
- net.rim.device.api.crypto.oid.OIDs.pbeWithSHAAnd40BitRC2_CBC
- net.rim.device.api.crypto.oid.OIDs.pbeWithSHAAnd40BitRC4
- net.rim.device.api.crypto.oid.OIDs.pkcs12_certBag
- net.rim.device.api.crypto.oid.OIDs.pkcs12_crlBag
- net.rim.device.api.crypto.oid.OIDs.pkcs12_keyBag
- net.rim.device.api.crypto.oid.OIDs.pkcs12_pkcs8ShroudedKeyBag
- net.rim.device.api.crypto.oid.OIDs.pkcs12_safeContentsBag
- net.rim.device.api.crypto.oid.OIDs.pkcs12_secretBag
- net.rim.device.api.crypto.oid.OIDs.pkcs9_sdsiCertificate
- net.rim.device.api.crypto.oid.OIDs.pkcs9_x509CRL
- net.rim.device.api.crypto.oid.OIDs.pkcs9_x509Certificate
- net.rim.device.api.i18n.Locale.LOCALE_ar
- net.rim.device.api.i18n.Locale.LOCALE_ca
- net.rim.device.api.i18n.Locale.LOCALE_cs
- net.rim.device.api.i18n.Locale.LOCALE_he
- net.rim.device.api.i18n.Locale.LOCALE_hu
- net.rim.device.api.i18n.Locale.LOCALE_ja
- net.rim.device.api.i18n.Locale.LOCALE_ko
- net.rim.device.api.io.ConnEvent.GM_RD_CRYPTO_SRC_KEY_MISMATCH
- net.rim.device.api.io.ConnEvent.GM_RH_NO_DEST
- net.rim.device.api.io.DatagramStatusListener.GME_ROUTE_FAILURE
- net.rim.device.api.io.MIMETypeAssociations.MEDIA_TYPE_PLAY_LIST
- net.rim.device.api.itpolicy.ITPolicy.GUID_PASSWORD_SET_BY_ITADMIN
- net.rim.device.api.ldap.LDAPQuery.LDAP_CONN_DEFAULT
- net.rim.device.api.ldap.LDAPQuery.LDAP_CONN_SSL_OR_TLS
- net.rim.device.api.ldap.LDAPQuery.LDAP_ERROR_BAD_DEFAULT_AUTHENTICATION_EXCEPTION
- net.rim.device.api.ldap.LDAPQuery.LDAP_ERROR_SSL_HANDSHAKE_EXCEPTION
- net.rim.device.api.ldap.LDAPQuery.setConnectionType(int)
- net.rim.device.api.math.Fixed32.divtoInt(int, int)
- net.rim.device.api.media.control.AudioPathControl.AUDIO_PATH_BLUETOOTH_A2DP
- net.rim.device.api.media.control.AuthenticationControl.setCredentials(java.lang.String)

© 2007 Research In Motion Limited. All rights reserved.

- net.rim.device.api.notification.NotificationsConstants.CONTINUOUS_REPEAT
- net.rim.device.api.servicebook.ServiceBook.getRecordByCidAndSid(java.lang.String, long)
- net.rim.device.api.synchronization.ConverterUtilities.readLongArray(net.rim.device.api.util.DataBuffer)
- net.rim.device.api.synchronization.ConverterUtilities.writeLongArray(net.rim.device.api.util.DataBuffer, int, long[])
- net.rim.device.api.synchronization.SyncManager.getLocalizedCollectionName(long, java.lang.String, net.rim.device.api.i18n.Locale)
- net.rim.device.api.system.Characters.IDEOGRAPHIC_SPACE
- net.rim.device.api.system.DeviceInfo.canResetIdleTime()
- net.rim.device.api.system.DeviceInfo.getLockTimeout()
- net.rim.device.api.system.DeviceInfo.getSoftwareVersion()
- net.rim.device.api.system.Display.screenshot(net.rim.device.api.system.Bitmap)
- net.rim.device.api.system.Display.screenshot(net.rim.device.api.system.Bitmap, int, int, int, int)
- net.rim.device.api.system.EventInjector.KeyCodeEvent.getStatus()
- net.rim.device.api.system.EventInjector.KeyCodeEvent.setStatus(int)
- net.rim.device.api.system.GPRSInfo.imeisvToString(byte[], boolean)
- net.rim.device.api.system.GPRSInfo.GPRSCellInfo.getMCC()
- net.rim.device.api.system.GPRSInfo.GPRSCellInfo.getMNC()
- net.rim.device.api.system.GPRSInfo.GPRSCellInfo.getRSSI()
- net.rim.device.api.system.JPEGEncodedImage.encode(net.rim.device.api.system.Bitmap, int)
- net.rim.device.api.system.LED.isSupported(int)
- net.rim.device.api.system.PNGEncodedImage.encode(net.rim.device.api.system.Bitmap)
- net.rim.device.api.system.RadioInfo.getEnabledWAFs()
- net.rim.device.api.system.SMSParameters.MESSAGE_CODING_KOREAN_KSX1001
- net.rim.device.api.system.WLANConnectionListener.WLAN_ASSOCIATION_FAILED
- net.rim.device.api.system.WLANConnectionListener.WLAN_AUTH_FAIL
- net.rim.device.api.system.WLANConnectionListener.WLAN_CONNECTION_TERMINATED
- net.rim.device.api.system.WLANConnectionListener.WLAN_DHCP_UNREACHABLE
- net.rim.device.api.system.WLANConnectionListener.WLAN_IP_ADDRESS_LOST
- net.rim.device.api.system.WLANConnectionListener.WLAN_OUT_OF_COVERAGE
- net.rim.device.api.system.WLANConnectionListener.WLAN_UNSPECIFIED_REASON
- net.rim.device.api.system.WLANConnectionListener.networkConnected()
- net.rim.device.api.system.WLANConnectionListener.networkDisconnected(int)
- net.rim.device.api.system.WLANInfo.WLAN_BAND_A
- net.rim.device.api.system.WLANInfo.WLAN_BAND_B
- net.rim.device.api.system.WLANInfo.WLAN_BAND_BG
- net.rim.device.api.system.WLANInfo.WLAN_STATE_CONNECTED
- net.rim.device.api.system.WLANInfo.WLAN_STATE_DISCONNECTED

© 2007 Research In Motion Limited. All rights reserved.

- net.rim.device.api.system.WLANInfo.addListener(net.rim.device.api.system.WLANListener)
- net.rim.device.api.system.WLANInfo.getAPInfo()
- net.rim.device.api.system.WLANInfo.getWLANState()
- net.rim.device.api.system.WLANInfo.removeListener(net.rim.device.api.system.WLANListener)
- net.rim.device.api.system.WLANInfo.WLANAPInfo.getBSSID()
- net.rim.device.api.system.WLANInfo.WLANAPInfo.getDataRate()
- net.rim.device.api.system.WLANInfo.WLANAPInfo.getProfileName()
- net.rim.device.api.system.WLANInfo.WLANAPInfo.getRadioBand()
- net.rim.device.api.system.WLANInfo.WLANAPInfo.getSSID()
- net.rim.device.api.system.WLANInfo.WLANAPInfo.getSignalLevel()
- net.rim.device.api.ui.Field.NON_SPELLCHECKABLE
- net.rim.device.api.ui.Field.SPELLCHECKABLE
- net.rim.device.api.ui.Field.SPELLCHECKABLE_MASK
- net.rim.device.api.ui.Field.isSpellCheckable()
- net.rim.device.api.ui.Field.setNonSpellCheckable(boolean)
- net.rim.device.api.ui.Keypad.HW_LAYOUT_39
- net.rim.device.api.ui.Screen.addScreenUiEngineAttachedListener(net.rim.device.api.ui.ScreenUiEngineAttachedListener)
- net.rim.device.api.ui.Screen.onUiEngineAttached(boolean)
- net.rim.device.api.ui.Screen.removeScreenUiEngineAttachedListener(net.rim.device.api.ui.ScreenUiEngineAttachedListener)
- net.rim.device.api.ui.ScreenUiEngineAttachedListener.onScreenUiEngineAttached(net.rim.device.api.ui.Screen, boolean)
- net.rim.device.api.ui.component.ActiveAutoTextEditField.fieldChangeNotify(int)
- net.rim.device.api.ui.component.AutoTextEditField.AUTOCAP_OFF
- net.rim.device.api.ui.component.AutoTextEditField.AUTOPERIOD_OFF
- net.rim.device.api.ui.component.AutoTextEditField.AUTOREPLACE_OFF
- net.rim.device.api.ui.component.Dialog.onUiEngineAttached(boolean)
- net.rim.device.api.ui.component.RadioButtonField.trackwheelClick(int, int)
- net.rim.device.api.ui.component.TextField.fieldChangeNotify(int)
- net.rim.device.api.ui.container.DialogFieldManager.deleteCustomFields()
- net.rim.device.api.util.Arrays.sort(byte[], int, int, char[])
- net.rim.device.api.util.DataBuffer.writeByteArray(byte[], int, int, boolean)
- net.rim.device.api.util.LongHashtable.keysToArray(long[])
- net.rim.device.api.xml.jaxp.DOMInternalRepresentation.setNamespaceAware(boolean)
- net.rim.device.api.xml.jaxp.RIMSAXParser.RIMSAXParser()
- net.rim.device.api.xml.jaxp.RIMSAXParser.getAllowUndefinedNamespaces()
- net.rim.device.api.xml.jaxp.RIMSAXParser.isNamespaceAware()
- net.rim.device.api.xml.jaxp.RIMSAXParser.isValidating()

© 2007 Research In Motion Limited. All rights reserved.

- net.rim.device.api.xml.jaxp.RIMSAXParser.parse(java.io.InputStream, org.xml.sax.helpers.DefaultHandler)
- net.rim.device.api.xml.jaxp.RIMSAXParser.parse(java.io.InputStream, org.xml.sax.helpers.DefaultHandler, java.lang.String[], java.lang.String[], java.lang.String[])
- net.rim.device.api.xml.jaxp.RIMSAXParser.parse(java.io.InputStream, org.xml.sax.helpers.DefaultHandler, net.rim.device.api.util.IntHashtable, net.rim.device.api.util.IntHashtable, net.rim.device.api.util.IntHashtable)
- net.rim.device.api.xml.jaxp.RIMSAXParser.parse(java.lang.String, org.xml.sax.helpers.DefaultHandler, boolean)
- net.rim.device.api.xml.jaxp.RIMSAXParser.parse(org.xml.sax.InputSource, org.xml.sax.helpers.DefaultHandler)
- net.rim.device.api.xml.jaxp.RIMSAXParser.setAllowUndefinedNamespaces(boolean)
- net.rim.device.api.xml.jaxp.RIMSAXParser.setNamespaceAware(boolean)
- net.rim.device.api.xml.jaxp.SAXParserImpl.setNamespaceAware(boolean)
- net.rim.device.api.xml.jaxp.XMLParser.setNamespaceAware(boolean)
- net.rim.device.api.xml.parsers.SAXParser.setNamespaceAware(boolean)
- org.w3c.dom.Attr.isId()
- org.xml.sax.ContentHandler.characters(char[], int, int)
- org.xml.sax.ContentHandler.endDocument()
- org.xml.sax.ContentHandler.endElement(java.lang.String, java.lang.String, java.lang.String)
- org.xml.sax.ContentHandler.endPrefixMapping(java.lang.String)
- org.xml.sax.ContentHandler.ignorableWhitespace(char[], int, int)
- org.xml.sax.ContentHandler.processingInstruction(java.lang.String, java.lang.String)
- org.xml.sax.ContentHandler.setDocumentLocator(org.xml.sax.Locator)
- org.xml.sax.ContentHandler.skippedEntity(java.lang.String)
- org.xml.sax.ContentHandler.startDocument()
- org.xml.sax.ContentHandler.startElement(java.lang.String, java.lang.String, java.lang.String, org.xml.sax.Attributes)
- org.xml.sax.ContentHandler.startPrefixMapping(java.lang.String, java.lang.String)
- org.xml.sax.DTDHandler.notationDecl(java.lang.String, java.lang.String, java.lang.String)
- org.xml.sax.DTDHandler.unparsedEntityDecl(java.lang.String, java.lang.String, java.lang.String, java.lang.String)
- org.xml.sax.SAXException.SAXException()
- org.xml.sax.SAXException.SAXException(java.lang.Exception)
- org.xml.sax.SAXException.SAXException(java.lang.String, java.lang.Exception)
- org.xml.sax.SAXException.getException()
- org.xml.sax.SAXException.getMessage()
- org.xml.sax.SAXException.toString()
- org.xml.sax.SAXNotRecognizedException.SAXNotRecognizedException()
- org.xml.sax.SAXNotSupportedException.SAXNotSupportedException()

© 2007 Research In Motion Limited. All rights reserved.

- org.xml.sax.SAXParseException.SAXParseException(java.lang.String, java.lang.String, java.lang.String, int, int)
- org.xml.sax.SAXParseException.SAXParseException(java.lang.String, java.lang.String, java.lang.String, int, int, java.lang.Exception)
- org.xml.sax.SAXParseException.SAXParseException(java.lang.String, org.xml.sax.Locator, java.lang.Exception)
- org.xml.sax.XMLFilter.getParent()
- org.xml.sax.XMLFilter.setParent(org.xml.sax.XMLReader)
- org.xml.sax.XMLReader.getContentHandler()
- org.xml.sax.XMLReader.getDTDHandler()
- org.xml.sax.XMLReader.getEntityResolver()
- org.xml.sax.XMLReader.getErrorHandler()
- org.xml.sax.XMLReader.getFeature(java.lang.String)
- org.xml.sax.XMLReader.getProperty(java.lang.String)
- org.xml.sax.XMLReader.parse(java.lang.String)
- org.xml.sax.XMLReader.parse(org.xml.sax.InputSource)
- org.xml.sax.XMLReader.setContentHandler(org.xml.sax.ContentHandler)
- org.xml.sax.XMLReader.setDTDHandler(org.xml.sax.DTDHandler)
- org.xml.sax.XMLReader.setEntityResolver(org.xml.sax.EntityResolver)
- org.xml.sax.XMLReader.setErrorHandler(org.xml.sax.ErrorHandler)
- org.xml.sax.XMLReader.setFeature(java.lang.String, boolean)
- org.xml.sax.XMLReader.setProperty(java.lang.String, java.lang.Object)
- org.xml.sax.helpers.AttributesImpl.AttributesImpl()
- org.xml.sax.helpers.AttributesImpl.AttributesImpl(org.xml.sax.Attributes)
- org.xml.sax.helpers.AttributesImpl.addAttribute(java.lang.String, java.lang.String, java.lang.String, java.lang.String, java.lang.String)
- org.xml.sax.helpers.AttributesImpl.clear()
- org.xml.sax.helpers.AttributesImpl.getIndex(java.lang.String)
- org.xml.sax.helpers.AttributesImpl.getIndex(java.lang.String, java.lang.String)
- org.xml.sax.helpers.AttributesImpl.getLength()
- org.xml.sax.helpers.AttributesImpl.getLocalName(int)
- org.xml.sax.helpers.AttributesImpl.getQName(int)
- org.xml.sax.helpers.AttributesImpl.getType(int)
- org.xml.sax.helpers.AttributesImpl.getType(java.lang.String)
- org.xml.sax.helpers.AttributesImpl.getType(java.lang.String, java.lang.String)
- org.xml.sax.helpers.AttributesImpl.getURI(int)
- org.xml.sax.helpers.AttributesImpl.getValue(int)
- org.xml.sax.helpers.AttributesImpl.getValue(java.lang.String)
- org.xml.sax.helpers.AttributesImpl.getValue(java.lang.String, java.lang.String)

© 2007 Research In Motion Limited. All rights reserved.

- org.xml.sax.helpers.AttributesImpl.removeAttribute(int)
- org.xml.sax.helpers.AttributesImpl.setAttribute(int, java.lang.String, java.lang.String, java.lang.String, java.lang.String, java.lang.String)
- org.xml.sax.helpers.AttributesImpl.setAttributes(org.xml.sax.Attributes)
- org.xml.sax.helpers.AttributesImpl.setLocalName(int, java.lang.String)
- org.xml.sax.helpers.AttributesImpl.setQName(int, java.lang.String)
- org.xml.sax.helpers.AttributesImpl.setType(int, java.lang.String)
- org.xml.sax.helpers.AttributesImpl.setURI(int, java.lang.String)
- org.xml.sax.helpers.AttributesImpl.setValue(int, java.lang.String)
- org.xml.sax.helpers.LocatorImpl.LocatorImpl()
- org.xml.sax.helpers.LocatorImpl.LocatorImpl(org.xml.sax.Locator)
- org.xml.sax.helpers.LocatorImpl.getColumnNumber()
- org.xml.sax.helpers.LocatorImpl.getLineNumber()
- org.xml.sax.helpers.LocatorImpl.getPublicId()
- org.xml.sax.helpers.LocatorImpl.getSystemId()
- org.xml.sax.helpers.LocatorImpl.setColumnNumber(int)
- org.xml.sax.helpers.LocatorImpl.setLineNumber(int)
- org.xml.sax.helpers.LocatorImpl.setPublicId(java.lang.String)
- org.xml.sax.helpers.LocatorImpl.setSystemId(java.lang.String)
- org.xml.sax.helpers.NamespaceSupport.NSDECL
- org.xml.sax.helpers.NamespaceSupport.XMLNS
- org.xml.sax.helpers.NamespaceSupport.NamespaceSupport()
- org.xml.sax.helpers.NamespaceSupport.declarePrefix(java.lang.String, java.lang.String)
- org.xml.sax.helpers.NamespaceSupport.getDeclaredPrefixes()
- org.xml.sax.helpers.NamespaceSupport.getPrefix(java.lang.String)
- org.xml.sax.helpers.NamespaceSupport.getPrefixes()
- org.xml.sax.helpers.NamespaceSupport.getPrefixes(java.lang.String)
- org.xml.sax.helpers.NamespaceSupport.getURI(java.lang.String)
- org.xml.sax.helpers.NamespaceSupport.isNamespaceDeclUris()
- org.xml.sax.helpers.NamespaceSupport.popContext()
- org.xml.sax.helpers.NamespaceSupport.processName(java.lang.String, java.lang.String[], boolean)
- org.xml.sax.helpers.NamespaceSupport.pushContext()
- org.xml.sax.helpers.NamespaceSupport.reset()
- org.xml.sax.helpers.NamespaceSupport.setNamespaceDeclUris(boolean)
- org.xml.sax.helpers.XMLFilterImpl.XMLFilterImpl()
- org.xml.sax.helpers.XMLFilterImpl.XMLFilterImpl(org.xml.sax.XMLReader)
- org.xml.sax.helpers.XMLFilterImpl.characters(char[], int, int)
- org.xml.sax.helpers.XMLFilterImpl.endDocument()

© 2007 Research In Motion Limited. All rights reserved.

- org.xml.sax.helpers.XMLFilterImpl.endElement(java.lang.String, java.lang.String, java.lang.String)
- org.xml.sax.helpers.XMLFilterImpl.endPrefixMapping(java.lang.String)
- org.xml.sax.helpers.XMLFilterImpl.error(org.xml.sax.SAXParseException)
- org.xml.sax.helpers.XMLFilterImpl.fatalError(org.xml.sax.SAXParseException)
- org.xml.sax.helpers.XMLFilterImpl.getContentHandler()
- org.xml.sax.helpers.XMLFilterImpl.getDTDHandler()
- org.xml.sax.helpers.XMLFilterImpl.getEntityResolver()
- org.xml.sax.helpers.XMLFilterImpl.getErrorHandler()
- org.xml.sax.helpers.XMLFilterImpl.getFeature(java.lang.String)
- org.xml.sax.helpers.XMLFilterImpl.getParent()
- org.xml.sax.helpers.XMLFilterImpl.getProperty(java.lang.String)
- org.xml.sax.helpers.XMLFilterImpl.ignorableWhitespace(char[], int, int)
- org.xml.sax.helpers.XMLFilterImpl.notationDecl(java.lang.String, java.lang.String, java.lang.String)
- org.xml.sax.helpers.XMLFilterImpl.parse(java.lang.String)
- org.xml.sax.helpers.XMLFilterImpl.parse(org.xml.sax.InputSource)
- org.xml.sax.helpers.XMLFilterImpl.processingInstruction(java.lang.String, java.lang.String)
- org.xml.sax.helpers.XMLFilterImpl.resolveEntity(java.lang.String, java.lang.String)
- org.xml.sax.helpers.XMLFilterImpl.setContentHandler(org.xml.sax.ContentHandler)
- org.xml.sax.helpers.XMLFilterImpl.setDTDHandler(org.xml.sax.DTDHandler)
- org.xml.sax.helpers.XMLFilterImpl.setDocumentLocator(org.xml.sax.Locator)
- org.xml.sax.helpers.XMLFilterImpl.setEntityResolver(org.xml.sax.EntityResolver)
- org.xml.sax.helpers.XMLFilterImpl.setErrorHandler(org.xml.sax.ErrorHandler)
- org.xml.sax.helpers.XMLFilterImpl.setFeature(java.lang.String, boolean)
- org.xml.sax.helpers.XMLFilterImpl.setParent(org.xml.sax.XMLReader)
- org.xml.sax.helpers.XMLFilterImpl.setProperty(java.lang.String, java.lang.Object)
- org.xml.sax.helpers.XMLFilterImpl.skippedEntity(java.lang.String)
- org.xml.sax.helpers.XMLFilterImpl.startDocument()
- org.xml.sax.helpers.XMLFilterImpl.startElement(java.lang.String, java.lang.String, java.lang.String, org.xml.sax.Attributes)
- org.xml.sax.helpers.XMLFilterImpl.startPrefixMapping(java.lang.String, java.lang.String)
- org.xml.sax.helpers.XMLFilterImpl.unparsedEntityDecl(java.lang.String, java.lang.String, java.lang.String, java.lang.String)
- org.xml.sax.helpers.XMLFilterImpl.warning(org.xml.sax.SAXParseException)
- org.xml.sax.helpers.XMLReaderFactory.createXMLReader()
- org.xml.sax.helpers.XMLReaderFactory.createXMLReader(java.lang.String)

Deprecated methods

- net.rim.blackberry.api.invoke.MemoArguments.ARG_VIEW

© 2007 Research In Motion Limited. All rights reserved.

- net.rim.device.api.system.Application.addPeripheralListener(net.rim.device.api.system.PeripheralListener)
- net.rim.device.api.system.Application.removePeripheralListener(net.rim.device.api.system.PeripheralListener)
- net.rim.device.api.system.Audio.AUDIO_CODEC_VOICENOTE
- net.rim.device.api.system.Audio.playFile(int, int, java.lang.String)
- net.rim.device.api.system.Audio.recordFile(int, int, java.lang.String)
- net.rim.device.api.system.Audio.stopFile(int, int, java.lang.String)
- net.rim.device.api.system.AudioFileListener.OPERATION_PLAY
- net.rim.device.api.system.AudioFileListener.OPERATION_RECORD
- net.rim.device.api.system.AudioFileListener.OPERATION_STOP
- net.rim.device.api.system.AudioFileListener.audioFileOperationComplete(int)
- net.rim.device.api.system.AudioFileListener.audioFileOperationFailed(int, int)
- net.rim.device.api.system.EncodedImage.setScale(int)
- net.rim.device.api.system.PeripheralListener.TYPE_CAR_CHARGER
- net.rim.device.api.system.PeripheralListener.TYPE_CRADLE
- net.rim.device.api.system.PeripheralListener.TYPE_NONE
- net.rim.device.api.system.PeripheralListener.TYPE_TRAVEL_CHARGER
- net.rim.device.api.system.PeripheralListener.TYPE_UNKNOWN
- net.rim.device.api.system.PeripheralListener.TYPE_USER
- net.rim.device.api.system.PeripheralListener.peripheralChange(int)
- net.rim.device.api.system.RadioStatusListener.CAUSE_NOT_APPLICABLE
- net.rim.device.api.system.RadioStatusListener.GMM_ATTACH_ACCEPT
- net.rim.device.api.system.RadioStatusListener.GMM_ATTACH_REJECT
- net.rim.device.api.system.RadioStatusListener.GMM_AUTH_CIPHER_REJECT
- net.rim.device.api.system.RadioStatusListener.GMM_AUTH_CIPHER_REQUEST
- net.rim.device.api.system.RadioStatusListener.GMM_GMM_INFORMATION
- net.rim.device.api.system.RadioStatusListener.GMM_IDENTITY_REQUEST
- net.rim.device.api.system.RadioStatusListener.GMM_MO_DETACH_ACCEPT
- net.rim.device.api.system.RadioStatusListener.GMM_MT_DETACH_REQUEST
- net.rim.device.api.system.RadioStatusListener.GMM_PTMSI_REALLOC_CMD
- net.rim.device.api.system.RadioStatusListener.GMM_RA_UPDATE_ACCEPT
- net.rim.device.api.system.RadioStatusListener.GMM_RA_UPDATE_REJECT
- net.rim.device.api.system.RadioStatusListener.MM_ABORT
- net.rim.device.api.system.RadioStatusListener.MM_AUTHENTICATION_REJECT
- net.rim.device.api.system.RadioStatusListener.MM_AUTHENTICATION_REQUEST
- net.rim.device.api.system.RadioStatusListener.MM_CM_SERVICE_ACCEPT
- net.rim.device.api.system.RadioStatusListener.MM_CM_SERVICE_REJECT

© 2007 Research In Motion Limited. All rights reserved.

- net.rim.device.api.system.RadioStatusListener.MM_IDENTITY_REQUEST
- net.rim.device.api.system.RadioStatusListener.MM_LOCATION_UPDATE_ACCEPT
- net.rim.device.api.system.RadioStatusListener.MM_LOCATION_UPDATE_REJECT
- net.rim.device.api.system.RadioStatusListener.MM_MM_INFORMATION
- net.rim.device.api.system.RadioStatusListener.MM_MM_STATUS
- net.rim.device.api.system.RadioStatusListener.MM_TMSI_REALLOC_CMD
- net.rim.device.api.system.RadioStatusListener.SM_DEACTIVATE_PDP_ACCEPT
- net.rim.device.api.system.RadioStatusListener.SM_DEACTIVATE_PDP_REQUEST
- net.rim.device.api.system.RadioStatusListener.SM_PDP_ACTIVATE_ACCEPT
- net.rim.device.api.system.RadioStatusListener.SM_PDP_ACTIVATE_REJECT
- net.rim.device.api.ui.Graphics.getScreenHeight()
- net.rim.device.api.ui.Graphics.getScreenHorizontalResolution()
- net.rim.device.api.ui.Graphics.getScreenVerticalResolution()
- net.rim.device.api.ui.Graphics.getScreenWidth()
- net.rim.device.api.ui.Screen.onDisplay()
- net.rim.device.api.ui.Screen.onUndisplay()
- net.rim.device.api.ui.component.ActiveRichTextField.getMainScreenManager()
- net.rim.device.api.ui.component.Dialog.Dialog(int, java.lang.String, int, net.rim.device.api.system.Bitmap, long, boolean)
- net.rim.device.api.ui.component.Dialog.getDontAskAgainValue()
- net.rim.device.api.ui.component.Dialog.onDisplay()
- net.rim.device.api.ui.component.Dialog.onUndisplay()
- net.rim.device.api.ui.component.Menu.setTarget(net.rim.device.api.ui.Field)

© 2007 Research In Motion Limited. All rights reserved.